In [ ]:
```
## Group No: 53

## Group Member Names:
| Name           | BITS ID        | Contribution |
|----------------|----------------|--------------|
| Mujtaba        | 2023ac05819    | 100%         |
| Kartik Batta   | 2023ac05422    | 100%         |
| Sachin Laddha  | 2023ac05564    | 100%         |
```

In [1]:
```
pip install gdown pandas matplotlib numpy
```

```
Requirement already satisfied: gdown in /Users/sachinladdha/anaconda3/lib/python3.1
2/site-packages (5.2.0)
Requirement already satisfied: pandas in /Users/sachinladdha/anaconda3/lib/python3.1
2/site-packages (2.2.2)
Requirement already satisfied: matplotlib in /Users/sachinladdha/anaconda3/lib/pytho
n3.12/site-packages (3.9.2)
Requirement already satisfied: numpy in /Users/sachinladdha/anaconda3/lib/python3.1
2/site-packages (1.26.4)
Requirement already satisfied: beautifulsoup4 in /Users/sachinladdha/anaconda3/lib/p
ython3.12/site-packages (from gdown) (4.12.3)
Requirement already satisfied: filelock in /Users/sachinladdha/anaconda3/lib/python
3.12/site-packages (from gdown) (3.13.1)
Requirement already satisfied: requests[socks] in /Users/sachinladdha/anaconda3/lib/
python3.12/site-packages (from gdown) (2.32.3)
Requirement already satisfied: tqdm in /Users/sachinladdha/anaconda3/lib/python3.12/
site-packages (from gdown) (4.66.5)
Requirement already satisfied: python-dateutil>=2.8.2 in /Users/sachinladdha/anacond
a3/lib/python3.12/site-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /Users/sachinladdha/anaconda3/lib/pyt
hon3.12/site-packages (from pandas) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in /Users/sachinladdha/anaconda3/lib/p
ython3.12/site-packages (from pandas) (2023.3)
Requirement already satisfied: contourpy>=1.0.1 in /Users/sachinladdha/anaconda3/li
b/python3.12/site-packages (from matplotlib) (1.2.0)
Requirement already satisfied: cycler>=0.10 in /Users/sachinladdha/anaconda3/lib/pyt
hon3.12/site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in /Users/sachinladdha/anaconda3/li
b/python3.12/site-packages (from matplotlib) (4.51.0)
Requirement already satisfied: kiwisolver>=1.3.1 in /Users/sachinladdha/anaconda3/li
b/python3.12/site-packages (from matplotlib) (1.4.4)
Requirement already satisfied: packaging>=20.0 in /Users/sachinladdha/anaconda3/lib/
python3.12/site-packages (from matplotlib) (24.1)
Requirement already satisfied: pillow>=8 in /Users/sachinladdha/anaconda3/lib/python
3.12/site-packages (from matplotlib) (10.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in /Users/sachinladdha/anaconda3/li
b/python3.12/site-packages (from matplotlib) (3.1.2)
Requirement already satisfied: six>=1.5 in /Users/sachinladdha/anaconda3/lib/python
3.12/site-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)
Requirement already satisfied: soupsieve>1.2 in /Users/sachinladdha/anaconda3/lib/py
thon3.12/site-packages (from beautifulsoup4->gdown) (2.5)
Requirement already satisfied: charset-normalizer<4,>=2 in /Users/sachinladdha/anaco
nda3/lib/python3.12/site-packages (from requests[socks]->gdown) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /Users/sachinladdha/anaconda3/lib/pyt
hon3.12/site-packages (from requests[socks]->gdown) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in /Users/sachinladdha/anaconda3/l
ib/python3.12/site-packages (from requests[socks]->gdown) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in /Users/sachinladdha/anaconda3/l
ib/python3.12/site-packages (from requests[socks]->gdown) (2024.8.30)
Requirement already satisfied: PySocks!=1.5.7,>=1.5.6 in /Users/sachinladdha/anacond
a3/lib/python3.12/site-packages (from requests[socks]->gdown) (1.7.1)
Note: you may need to restart the kernel to use updated packages.
```

In [2]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
from collections import defaultdict
import gdown
```

In [3]:
```
# Download the dataset from Google Drive
file_id = "1gfobhqlVCw8Oo52JCiYpEBGhG5k7cWBr"
url = f"https://drive.google.com/uc?id={file_id}&export=download"
dataset_path = "movie_ratings.csv"
gdown.download(url, dataset_path, quiet=False)
```

```
Downloading...
From: https://drive.google.com/uc?id=1gfobhqlVCw8Oo52JCiYpEBGhG5k7cWBr&export=downlo
ad
To: /Users/sachinladdha/study/BITS/Deep Reinforcement Learning/Assignment-1/movie_ra
tings.csv
100%|████████████████████████████████████████| 2.48M/2.48M [00:00<00:00, 4.11MB/s]
```

Out[3]:
```
'movie_ratings.csv'
```

In [4]:
```
# Load and preprocess the dataset (Step 1)
df = pd.read_csv(dataset_path)
```

In [5]:
```
# Binarize rewards: Rating >= 4 -> Reward = 1, else Reward = 0
df['Reward'] = (df['rating'] >= 4).astype(int)
```

In [6]:
```
# Define the environment and compute rewards (Step 2)
movies = df['movieId'].unique()
movie_rewards = {movie: [] for movie in movies}

for movie in movies:
    movie_rewards[movie] = df[df['movieId'] == movie]['Reward'].tolist()

def get_reward(movie_id):
    rewards = movie_rewards[movie_id]
    return rewards[np.random.randint(len(rewards))]
```

In [7]:
```
# Simulation setup
num_iterations = 1000
cumulative_rewards = defaultdict(list)
```

In [8]:
```
# Random Policy (Step 3)
def random_policy():
    total_reward = 0
    for _ in range(num_iterations):
        movie = np.random.choice(movies)
        reward = get_reward(movie)
        total_reward += reward
        cumulative_rewards['Random'].append(total_reward)
random_policy()
```

In [9]:
```
# Greedy Policy (Step 4)
def greedy_policy():
    total_reward = 0
    estimated_rewards = {movie: 0 for movie in movies}
    counts = {movie: 0 for movie in movies}
```

```
    for _ in range(num_iterations):
        if 0 in counts.values():  # If an arm hasn't been tried yet, try it
            movie = np.random.choice([m for m in movies if counts[m] == 0])
        else:
            movie = max(movies, key=lambda m: estimated_rewards[m])

        reward = get_reward(movie)
        counts[movie] += 1
        estimated_rewards[movie] += (reward - estimated_rewards[movie]) / counts[mo
        total_reward += reward
        cumulative_rewards['Greedy'].append(total_reward)
greedy_policy()
```

In [10]:
```
# Epsilon-Greedy Policy (Step 5)
def epsilon_greedy_policy(epsilon):
    total_reward = 0
    estimated_rewards = {movie: 0 for movie in movies}
    counts = {movie: 0 for movie in movies}

    for _ in range(num_iterations):
        if np.random.rand() < epsilon:  # Explore
            movie = np.random.choice(movies)
        else:  # Exploit
            movie = max(movies, key=lambda m: estimated_rewards[m])

        reward = get_reward(movie)
        counts[movie] += 1
        estimated_rewards[movie] += (reward - estimated_rewards[movie]) / counts[mo
        total_reward += reward
        cumulative_rewards[f'Epsilon-{epsilon}'].append(total_reward)

for epsilon in [0.1, 0.2, 0.5]:
    epsilon_greedy_policy(epsilon)
```

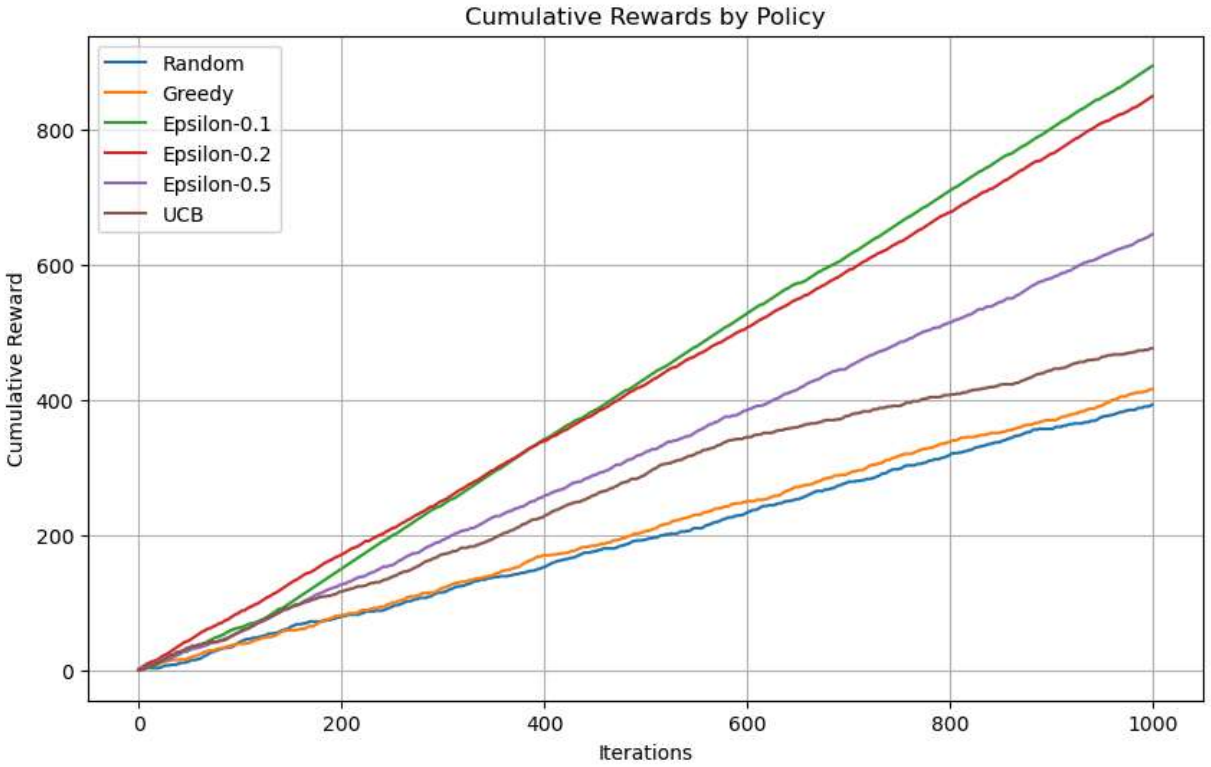
In [11]:
```
# UCB Algorithm (Step 6)
def ucb_policy():
    total_reward = 0
    estimated_rewards = {movie: 0 for movie in movies}
    counts = {movie: 0 for movie in movies}

    for t in range(1, num_iterations + 1):
        movie = max(movies, key=lambda m: estimated_rewards[m] + np.sqrt(2 * np.log
        reward = get_reward(movie)
        counts[movie] += 1
        estimated_rewards[movie] += (reward - estimated_rewards[movie]) / counts[mo
        total_reward += reward
        cumulative_rewards['UCB'].append(total_reward)
ucb_policy()
```

In [12]:
```
# Plot Cumulative Rewards (Step 7)
plt.figure(figsize=(10, 6))
for policy, rewards in cumulative_rewards.items():
    plt.plot(rewards, label=policy)

plt.xlabel('Iterations')
```

```
plt.ylabel('Cumulative Reward')
plt.title('Cumulative Rewards by Policy')
plt.legend()
plt.grid()
plt.show()
```



In [13]:
```
# Conclusion (Step 8)
print("\nConclusion:")
print("The policy with the highest cumulative reward is:", max(cumulative_rewards,
```

```
Conclusion:
The policy with the highest cumulative reward is: Epsilon-0.1
```

In [ ]: